

Университет ИТМО  
Факультет ПИиКТ

Системы искусственного интеллекта  
Лабораторная работа №4  
Нейронные сети

Выполнила: Наумова Н.А.

Группа Р33022

Преподаватель: Бессмертный И.А.

Санкт-Петербург

2020 г.

## Цель работы:

изучение гиперпараметров нейронной сети, понимание влияния различных параметров на точность обучения нейросети.

### Задание:

There are 2 parts of work in files named Lab1-Part1 and Lab1-Part2 respectively. Both parts represent work with collections of training and test data. Part 1 represent recognition of basic math functions with illustrations of neural net vision of the functions for training. Part 2 represent recognition of simple images for making the following work. Data usage represented at <https://keras.io/datasets/> and depends on variant.

There are represented such hyperparameters as

- Layer count
  - Neurons count per layer (actually it's not hyperparameter but structure parameter)
  - Learn rate
  - Regularization L1 and L2
  - Output layer activation type
  - Layer activation type
  - Loss function type
  - Epoch count
1. By changing these hyperparameters try to reach max accuracy value(at least 0.95) for Part2 model with fixed epoch count 20
  2. Change 1st hyperparameter's value from min to max with minimal step depends on your variant
  3. Show impact on result using graphs
  4. Describe impact of each hyperparameter on accuracy.
  5. Set hyperparameter value back to one which produced max accuracy
  6. Repeat 2-5 steps for second hyperparameter

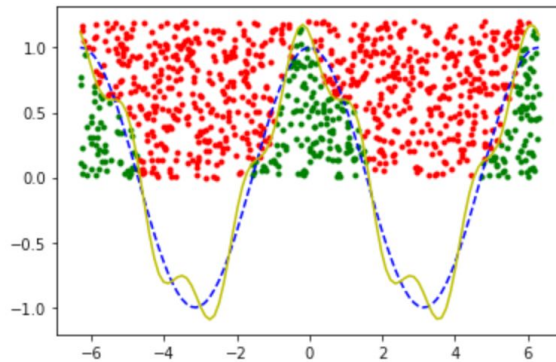
Var	Part1 func	Part2 data	Hyperparameters
1	Sin(x) X: -6,3..6.3 Y: 0..1.2	CIFAR10	Layers count, neurons count per layer
2	Cos(x) X: -9..9 Y: -1..1	CIFAR100	Learn rate, regularization L1
3	Absolute(Sin(x)) X: 6,3..6.3 Y: 0..1.2	Handwritten digits	Regularization L2, output layer activation type
4	Absolute(Cos(x)) X: -9..9 Y: -1..1	Fashion articles	Layer activation type, loss function type

# Выполнение ([https://github.com/mmmlpmsw/artificial\\_intelligence\\_systems/tree/master/lab4](https://github.com/mmmlpmsw/artificial_intelligence_systems/tree/master/lab4)):

## Часть 1:

### График функции

[<matplotlib.lines.Line2D at 0x7f8b4da399e8>]



Слоев: 3

Loss function: Binary crossentropy

Batch size: 97

Learn rate: -0.1, 0, 0.001, 0.01, 0.05, 0.1

Regularization L1: -0.1, 0, 0.0001, 0.0005, 0.001

Regularization L2: -0.1, 0, 0.0001, 0.0005, 0.001

Output layer activation type: sigmoid

Epoch count: 1190

Neurons count in layer 1: 10

Neurons count in layer 2: 8

Neurons count in layer 1: 10

Neurons count in layer 1: 0

Layer 1 activation type: relu

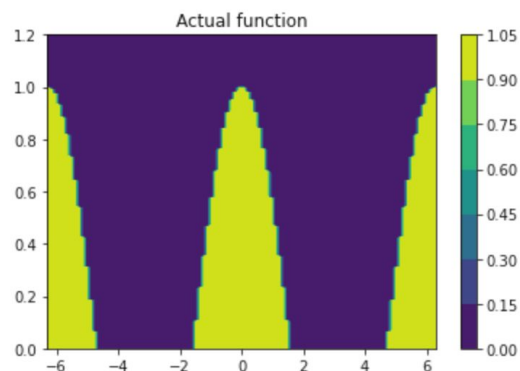
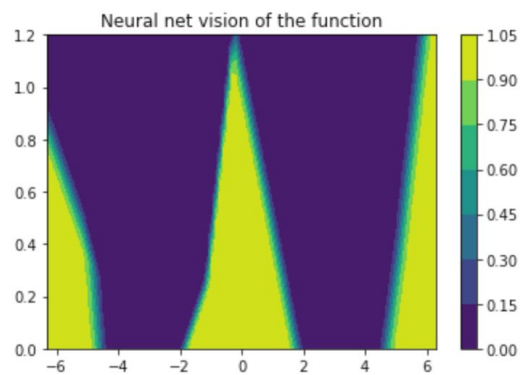
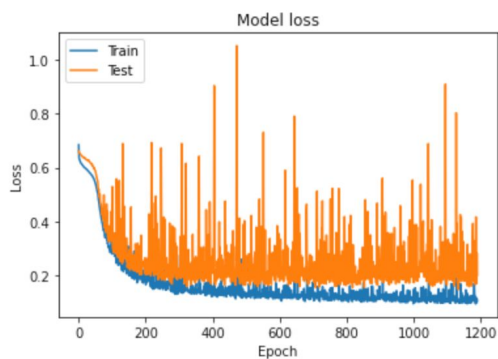
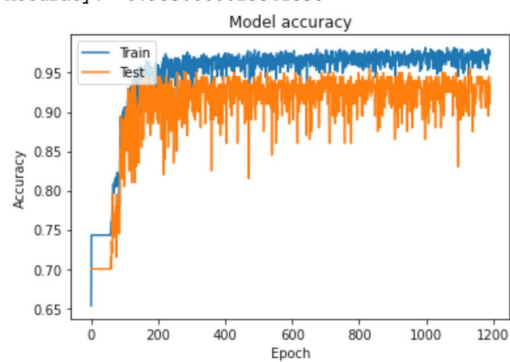
Layer 2 activation type: relu

Layer 3 activation type: relu

Layer 4 activation type: relu

Run Interact

Accuracy: 0.9350000023841858



## Часть 2

Меняем коли-во слоев.

Layers:  1

Loss function:

Batch size:

Learn rate:  (Options: -0.1, 0, 0.001, 0.01, 0.05, 0.1, 0.5, 1)

Regularization L1:  (Options: -0.1, 0, 0.0001, 0.0005, 0.001, 0.005, 0.01)

Regularization L2:  (Options: -0.1, 0, 0.0001, 0.0005, 0.001, 0.005, 0.01)

Output layer activation type:

Epoch count:  20

Neurons per layer:  6

Layer 1 activation type:

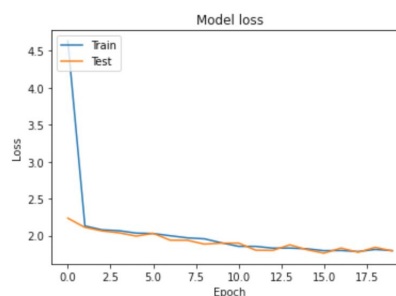
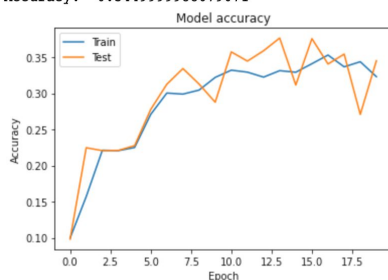
Layer 2 activation type:

Layer 3 activation type:

Layer 4 activation type:

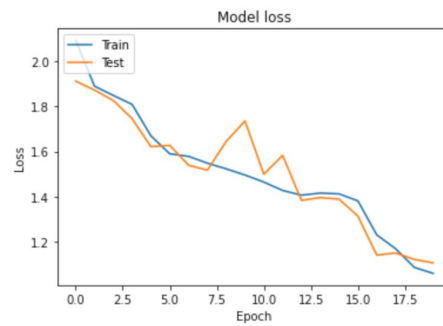
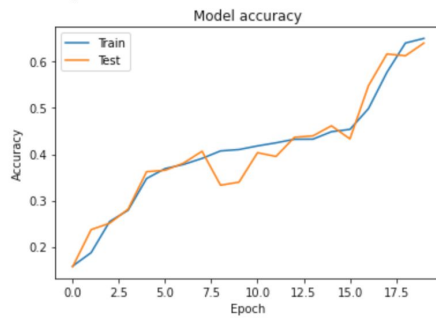
## 1 слой

Accuracy: 0.3449999988079071



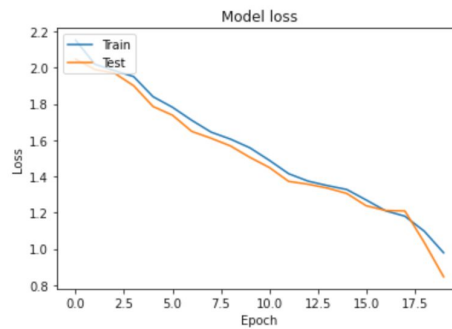
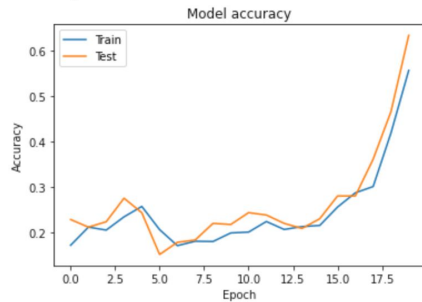
## 2 слоя

Accuracy: 0.6402999758720398



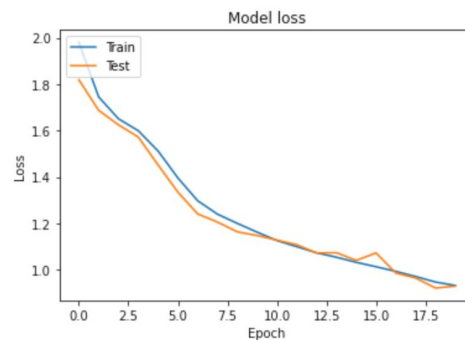
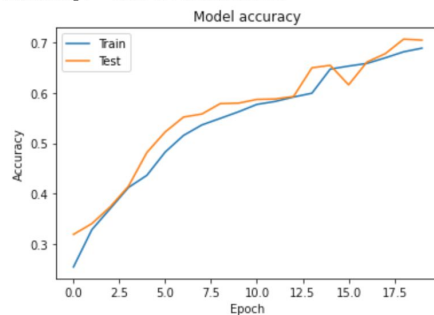
## 3 слоя

Accuracy: 0.6330999732017517



## 4 слоя

Accuracy: 0.7045000195503235

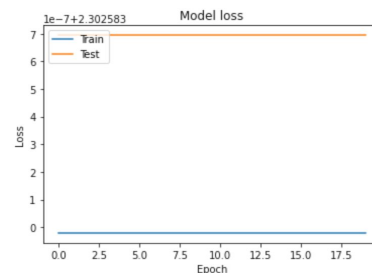
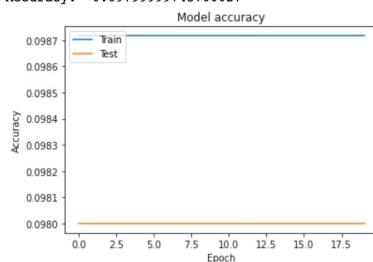


Из полученных графиков видим, что чем больше слоев в нейросети, тем выше ее точность.

Теперь будем менять количество нейронов на слое.

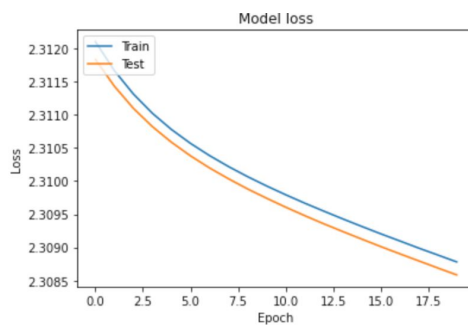
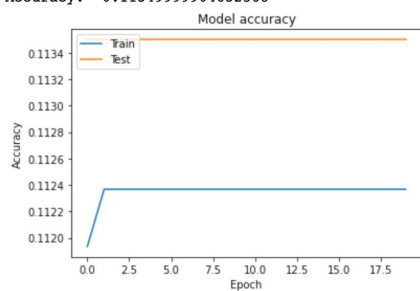
## 1 нейрон:

Accuracy: 0.09799999743700027



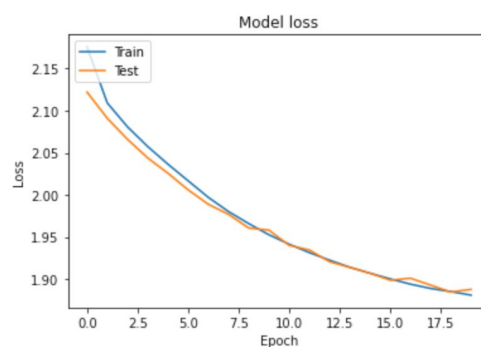
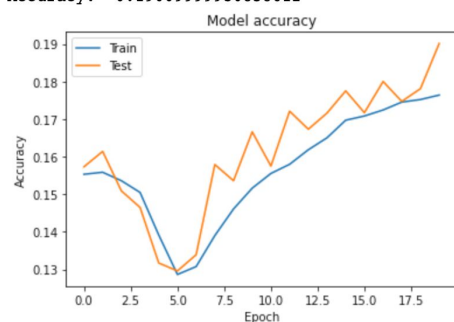
## 2 нейрона:

Accuracy: 0.11349999904632568



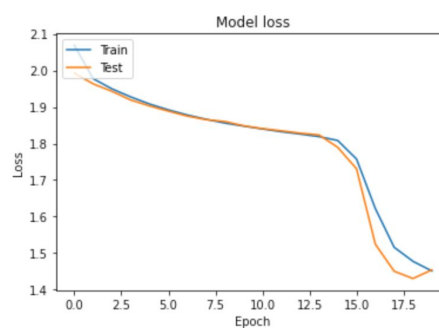
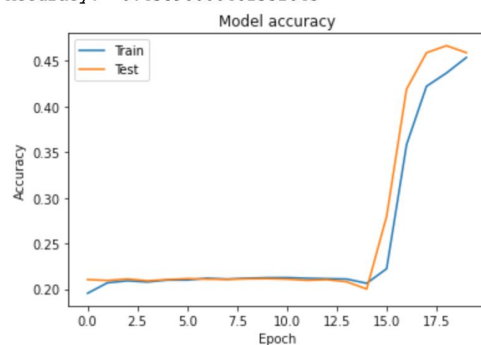
3 нейрона:

Accuracy: 0.19009999930858612



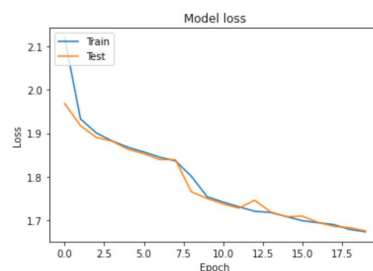
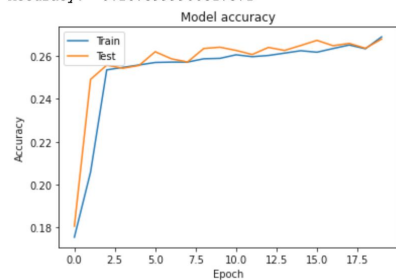
4 нейрона:

Accuracy: 0.45890000462532043



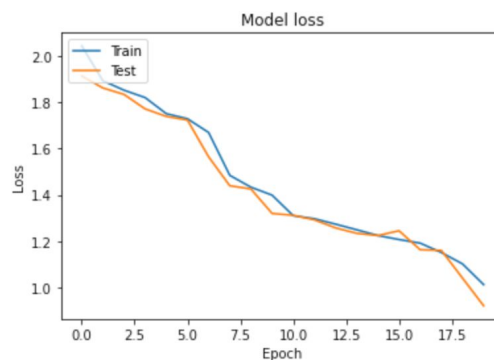
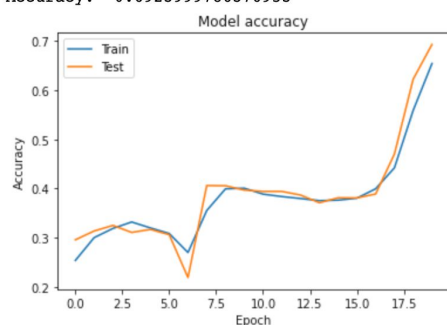
5 нейронов:

Accuracy: 0.2678999900817871



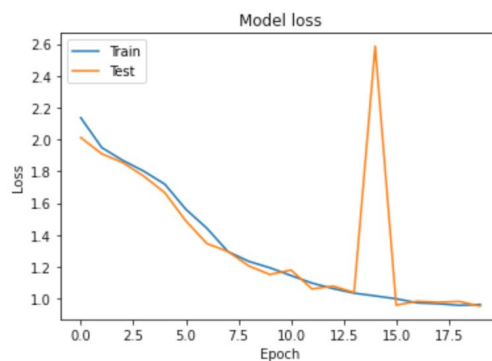
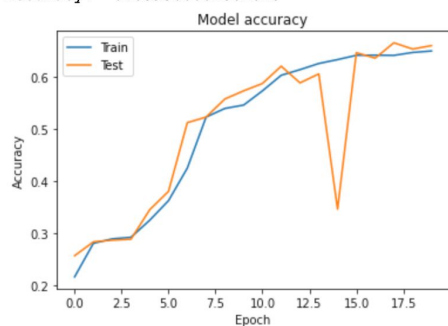
6 нейронов:

Accuracy: 0.6923999786376953



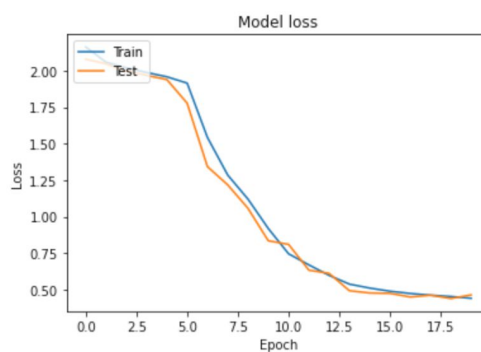
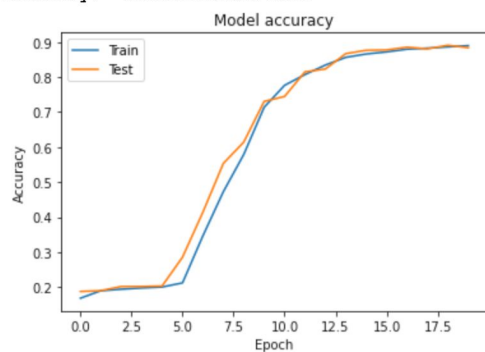
7 нейронов:

Accuracy: 0.659500002861023



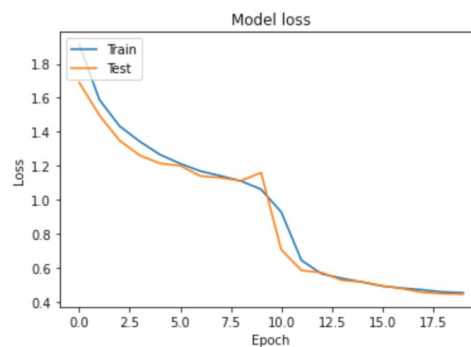
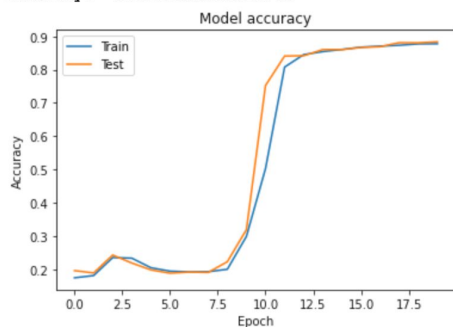
8 нейронов:

Accuracy: 0.8838000297546387



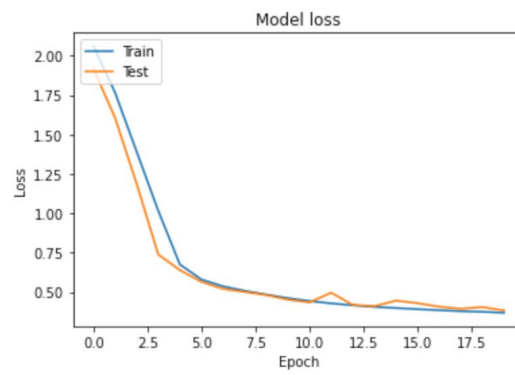
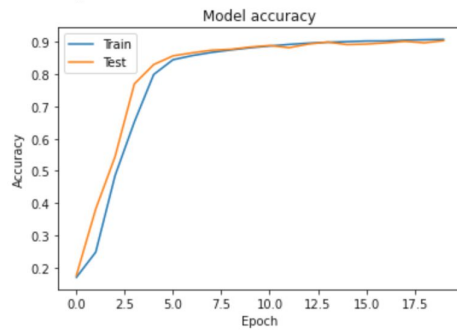
9 нейронов:

Accuracy: 0.8841000199317932



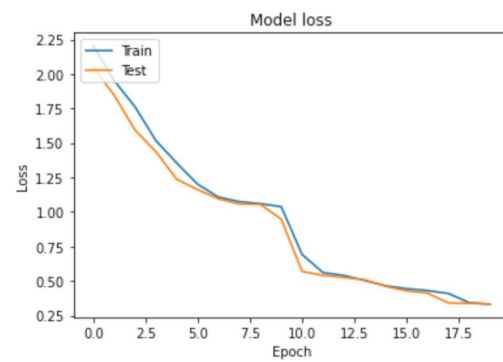
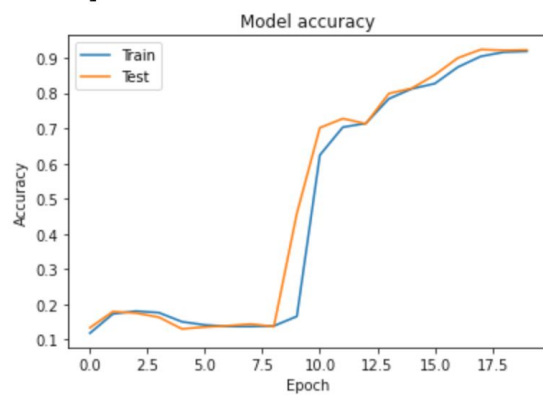
10 нейронов:

Accuracy: 0.904100008583069



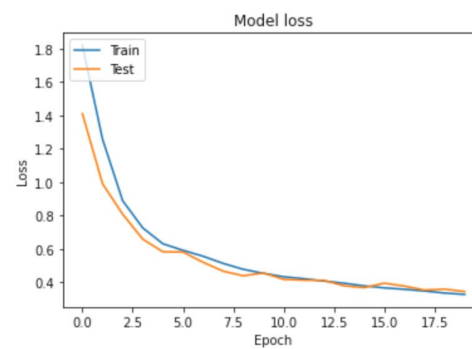
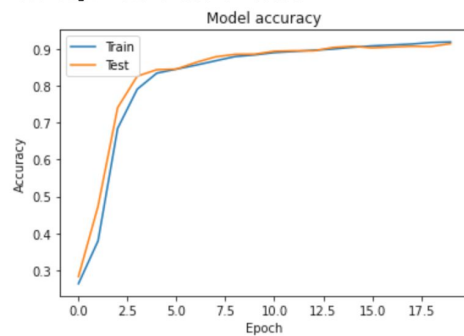
11 нейронов:

Accuracy: 0.9225000143051147



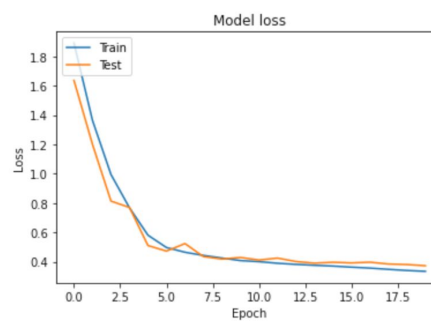
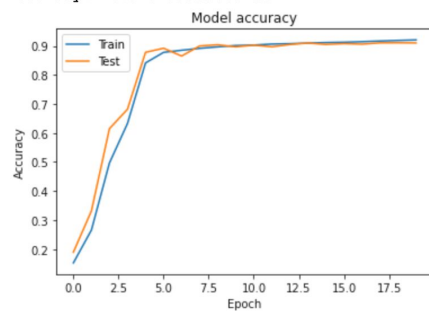
12 нейронов:

Accuracy: 0.9154000282287598



13 нейронов:

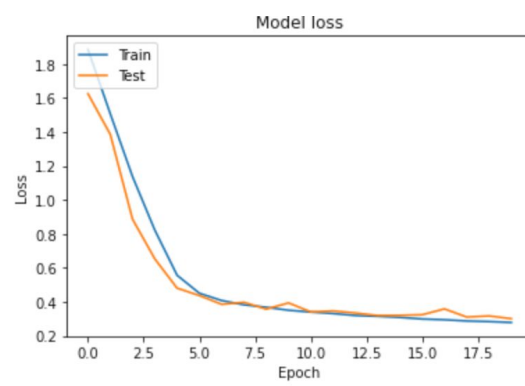
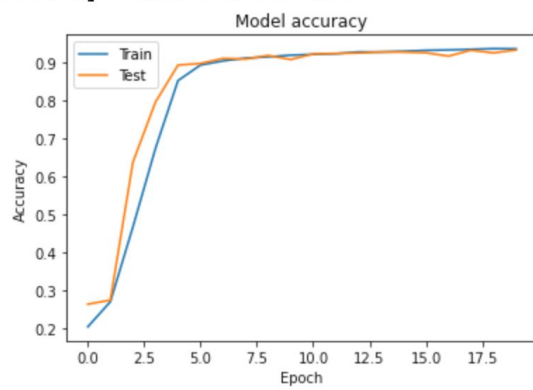
Accuracy: 0.910199998092651





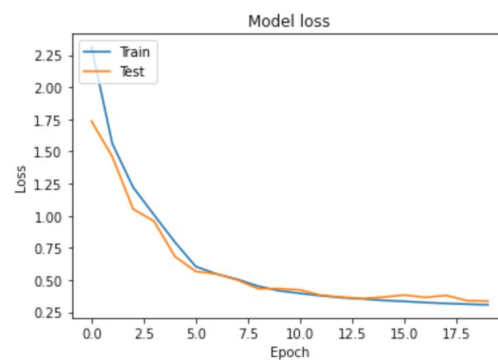
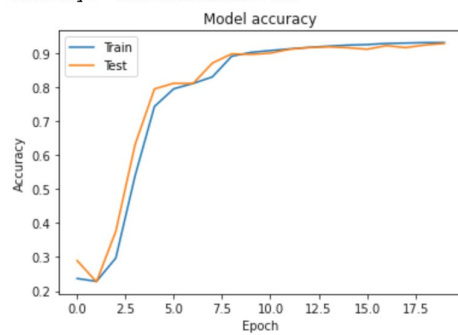
14 нейронов:

Accuracy: 0.9334999918937683



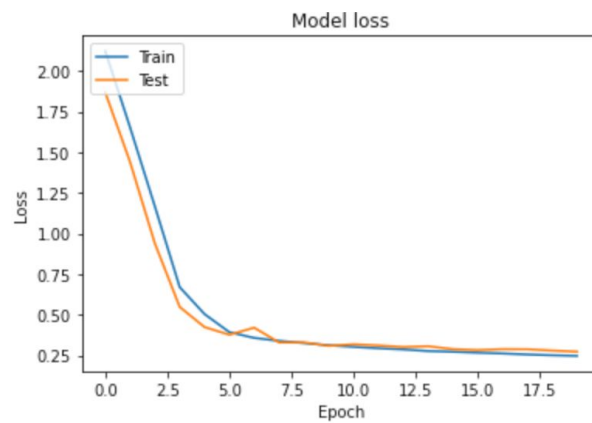
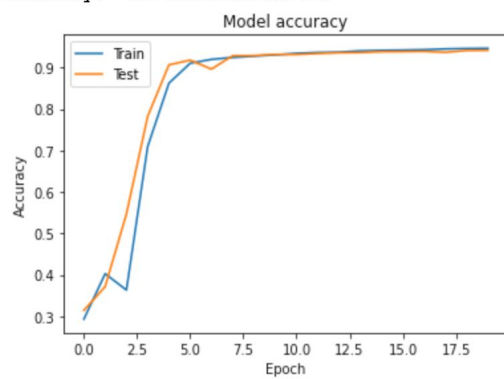
15 нейронов:

Accuracy: 0.9275000095367432



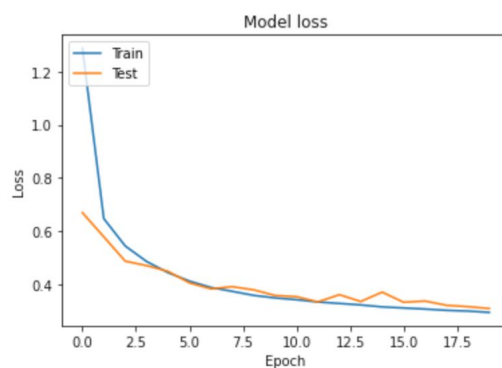
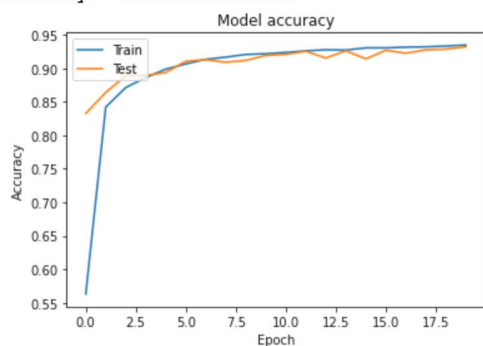
16 нейронов:

Accuracy: 0.9420999884605408



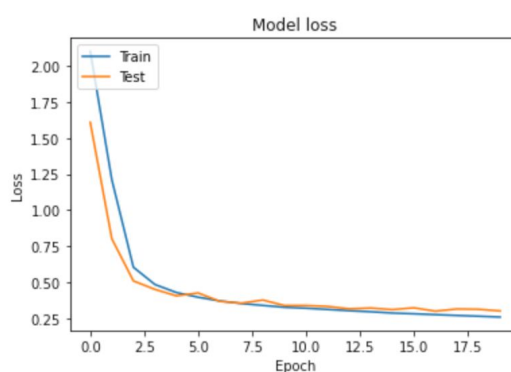
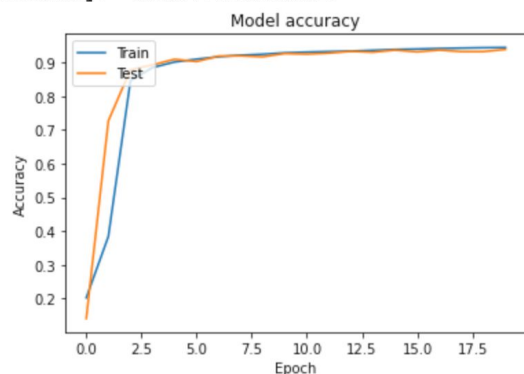
17 нейронов:

Accuracy: 0.932200014591217



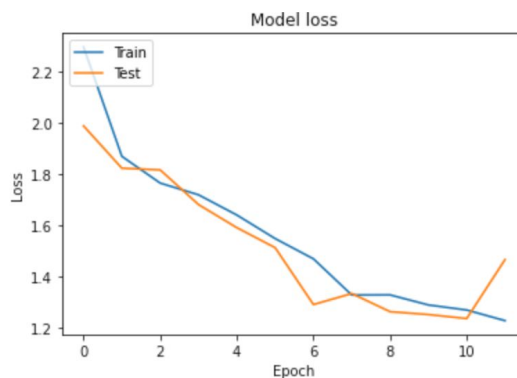
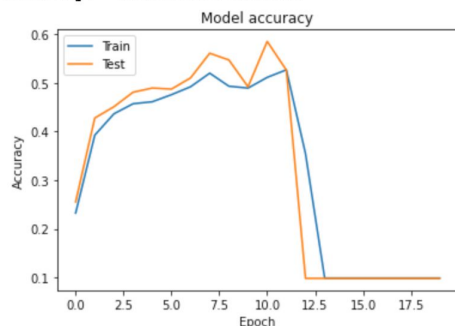
18 нейронов:

Accuracy: 0.9387000203132629



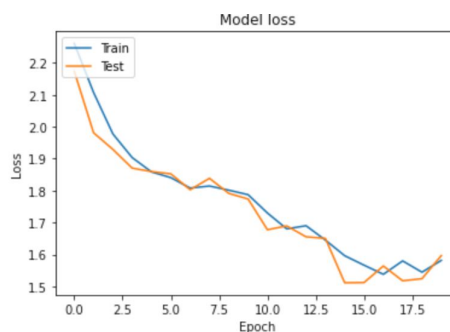
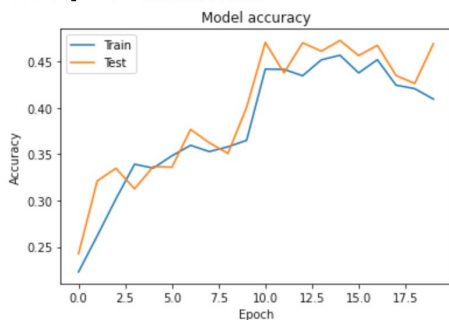
19 нейронов:

Accuracy: 0.09799999743700027



20 нейронов:

Accuracy: 0.46889999508857727



Промежуточный вывод: чем больше нейронов в слое, тем больше точность, но в то же время когда нейронов станет слишком много, то может произойти переобучение (как и произошло при 19 нейронах, когда точность очень резко упала).

## Параметры нейросети:

- Layer count - чем больше слоев, тем выше будет точность, однако не стоит перегружать нейросеть, иначе точность упадет
- Learn rate - это параметр градиентных алгоритмов обучения нейронных сетей, позволяющий управлять величиной коррекции весов на каждой итерации. Если скорость обучения слишком мала, то даже после обучения нейронной сети в течение длительного времени она будет далека от оптимальных результатов. С другой стороны, если скорость обучения слишком высока, то сеть очень быстро выдаст ответы.
- Regularization L1 and L2 (регуляризация)- это любая модификация, которую мы вносим в алгоритм обучения, предназначенный для уменьшения его ошибки обобщения. Регуляризация экономит ресурсы — зануляя ненужные веса, тем самым отключая нейроны (L1), а также “выпрямляет” функцию активации, приближая ее к линейной, тем самым позволяя избежать переобучения. На самом деле не выпрямляет, а сокращает ее до наиболее линейного отрезка функции активации.
- Layer activation type - определяет выходное значение нейрона в зависимости от результата взвешенной суммы входов и порогового значения. Она определяет, какие нейроны будут активированы и какая информация будет передаваться последующим слоям. Без функций активации глубокие сети теряют значительную часть своей способности к обучению. Нелинейность этих функций отвечает за повышение степени свободы, что позволяет обобщать проблемы высокой размерности в более низких измерениях.
- Loss function type (функция потерь) - мера, описывающая качество нейросети. Функция потерь измеряет «насколько хороша» нейронная сеть в отношении данной обучающей выборки и ожидаемых ответов. Она также может зависеть от таких переменных, как веса и смещения. Функция потерь одномерна и не является вектором, поскольку она оценивает, насколько хорошо нейронная сеть работает в целом.
- Epoch count - для увеличения точности нейронной сети количество эпох следует выбирать достаточно большим, главное — не допустить переобучения, которое происходит, если число эпох слишком большое.

Показатели, дающие очень высокую точность:

Layers:  3

Loss function:

Batch size:

Learn rate: 

-0.1	0	0.001	0.01	0.05	0.1	0.5	1
5							

Regularization L1: 

-0.1	0	0.0001	0.0005	0.001	0.005	0.01
0.05	0.1					

Regularization L2: 

-0.1	0	0.0001	0.0005	0.001	0.005	0.01
0.05	0.1					

Output layer activation type:

Epoch count:  20

Neurons per layer:  12

Layer 1 activation type:

Layer 2 activation type:

Layer 3 activation type:

Layer 4 activation type:

Вывод:

выполнив данную работу, я узнала больше о нейронных сетях и смогла реализовать свою собственную.